

# PreciseCam: Precise Camera Control for Text-to-Image Generation

## Supplementary Material

The supplementary material for *PreciseCam: Precise Camera Control for Text-to-Image Generation* includes this PDF document, an HTML browser featuring additional image results, and a demo video showing the usability of our model.

### A. Training Details

To train our framework we use the ControlNet [54] loss function. PF-US maps are encoded as RGB images, where the up-vector coordinates are scaled from  $[-1, 1]$  to  $[0, 255]$  and assigned to the R and G channels, while latitude values are mapped from  $[-90, 90]$  to  $[0, 255]$  and represented in the B channel. We initialize ControlNet using the SDXL model weights from Stability AI<sup>1</sup>, and train it on our entire dataset using the Adam optimizer [26] with hyperparameters  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , weight decay  $w = 10^{-2}$ , and a learning rate of  $10^{-6}$ . We employ a total batch size of 32, an input resolution of  $1024 \times 1024$  pixels, 16 floating-point precision, and 70,000 steps. As per usual practice, 50% of the text prompts are replaced by empty strings during training. The training was executed within the Accelerate framework [17] for four days on eight NVIDIA RTX A100 GPUs.

### B. Dataset Details

To train our model, we require triplets of RGB images, corresponding text prompts, and PF-US camera parameters ( $I_i, p_i, \Omega_i$ ). It is essential for our dataset to be diverse in both content and camera parameter values. We explore several approaches:

- *Existing Datasets:* Jin et al. [24] present a dataset containing RGB images paired with ground-truth PF maps. However, they primarily depict urban outdoor scenes and lack comprehensive coverage of camera parameters. For instance, images depicting large vertical FoVs or extreme distortions are absent.
- *PF estimators:* Previous works offer deep-learning models to estimate the PF map of a given image [24, 41], primarily intended for camera calibration. Thus, an alternative approach might be to apply this model to an existing image dataset, thus obtaining its associated PF maps. However, the estimated PF maps lack the precision required for our training needs, limiting our model’s ability to learn effective camera view control across the full range of camera parameters (see Fig. 13). Moreover, PF estimation models do not always consider the distortion parameter  $\xi$ .

<sup>1</sup><https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0>

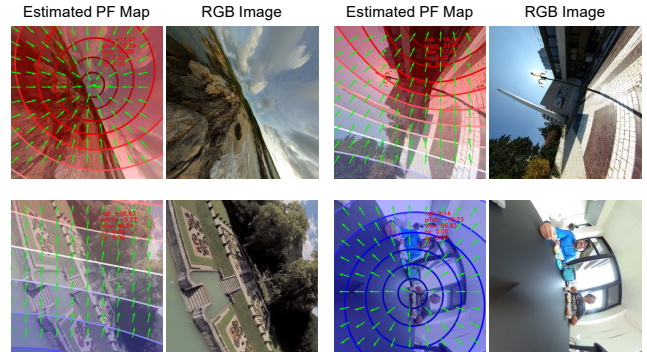


Figure 13. Incorrect PF map estimations using the model from Jin et al. [24] for different RGB images. These errors make the estimator unsuitable for our dataset creation, as the introduced noise is substantial enough to compromise our model’s training.

- *Cropping 360° images:* By using 360° images, we can extract patches corresponding to specific camera parameters, providing ground-truth PF-US maps that are crucial for our application. This approach allows us to sample the entire range of camera parameters while leveraging the Unified Spherical camera model, including its  $\xi$  distortion parameter.

To generate our dataset with ground-truth PF-US maps, we adopt this last approach using 360° images. We sample our set of camera parameters and obtain, for each sampled quartet  $\Omega = (\text{roll}, \text{pitch}, \text{vFoV}, \xi)$  the corresponding patches cropped from the 360° images, and their PF-US maps. To maximize content diversity, we use six different 360° image datasets: 360-SOD [28], CVRG-Pano [33], F-360iSOD [55], Poly Haven HDRIs [18], Sitzmann et al. [40], and 360cities [1]. These feature outdoor and indoor scenes, containing both natural and urban settings with diverse activities and environments.

From each 360° image, we sample 24 patches. To maximize the content diversity that each 360° image has to offer and avoid repeatedly sampling the same areas, the image is divided into six regions, with four patches sampled from each region using different camera parameters  $\Omega$ . For each region, we randomly sample yaw (necessary only to establish the 360° image horizontal coordinate) and pitch. For each pair of yaw and pitch, we randomly sample two vFoV values (one small  $\in (0, 0.5)$  and one large  $\in [0.5, 1)$ ), two  $\xi$  values (low  $\in [15, 60]$  and high  $\in [60, 140]$ ), yielding four possible combinations. We sample a roll rotation for each combination to generate four distinct image crops of the same region. This approach ensures that the same image content is depicted across different image crops, showc-

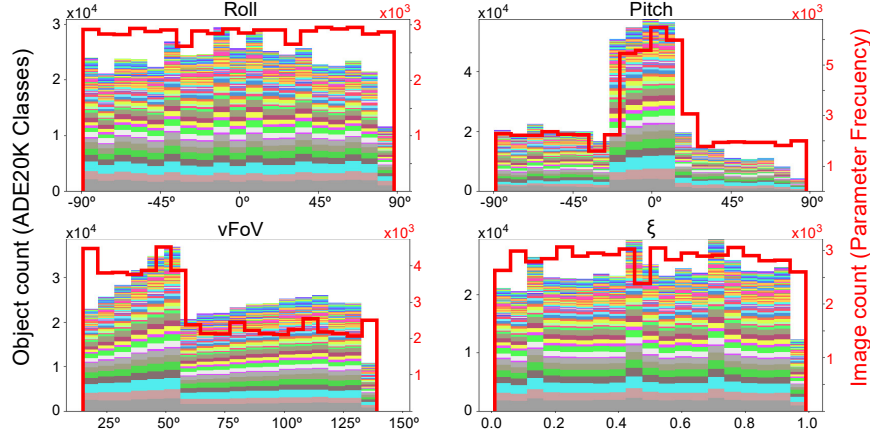


Figure 14. Object count for each ADE20K class per parameter range in our dataset (colored bars; each color represents a class), and parameter frequency distribution (red line) as the number of images for each value. Please note the different scales.

ing both minimum and maximum vFoVs and varying distortion levels at different rotations. This allows the model to learn how these parameters influence the final image content (e.g., how the appearance of a chair at a high vFoV varies when  $\xi$  is increased or decreased).

This results in a dataset of 57,380 RGB images with a ground-truth PF-US condition. Sampling ranges for each camera parameter are: roll  $\in (-90^\circ, 90^\circ)$ , pitch  $\in (-90^\circ, 90^\circ)$ , vFoV  $\in [15^\circ, 140^\circ]$  and  $\xi \in (0, 1)$ . We use BLIP-2 [29] to generate a descriptive text prompt  $p_i$  for each image  $I_i$ .

### B.1. Dataset Analysis

We ensure comprehensive camera parameter coverage by systematically selecting camera parameters for each image. As shown in Fig. 14, the dataset covers the full sampling range of parameters, though specific ranges of pitch and vFoV are more heavily represented. This design choice was aimed at increasing content variability: the greater representation of pitch values corresponds to the equator of panoramic images, as it contains the most semantic information [40], and the lower vFoV over-representation focuses on increasing the presence of first-plane objects from diverse camera angles while reducing image crop overlap.

Our dataset consists of images from six well-established datasets, covering a diverse range of scenes. To evaluate content diversity across camera parameter ranges, we used the Segformer [47] model to identify the number of distinct classes (visualized by different colors in Fig. 14).

## C. Additional Results

We present additional results of PreciseCam for various prompts in the form of an HTML browser. We show how our model can accurately generate images with the specified camera view. Within each tab, we display in each row the

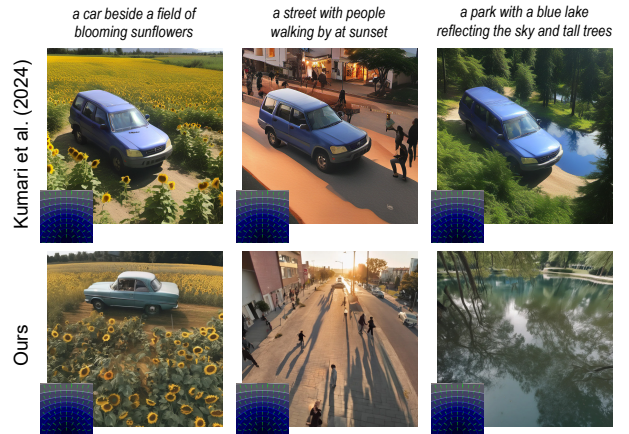


Figure 15. Kumari’s model [27] needs to be trained on specific objects (a car in this example), which will wrongly appear in the output despite being absent in the prompts.

generated images for the same prompt when a single camera parameter is varied while keeping the others fixed. The first row of each tab shows the PF-US for the corresponding camera settings. Note that the quality of the images has been reduced to meet the upload size in the paper submission platform.

Additionally, although Kumari et al. [27] address a different problem (i.e., controlling camera view while generating novel object views), we provide a comparison in Fig. 15, showing how it introduces the training object even when absent in the prompt.

## D. Prompt Engineering for Baseline SDXL

In Sec. 4 and Fig. 2, we show how our model maintains the text prompt adherence exhibited by the baseline SDXL despite the inclusion of camera control, achieving comparable

CLIP and BLIP scores [22]. This comparison is based on 2,940 images generated by both our method and the baseline SDXL. To enable the baseline SDXL to produce the correct camera views, we employed prompt engineering, explicitly specifying the desired view to encourage distinct camera perspectives.

This section outlines the prompt engineering techniques applied to SDXL. After extensive testing, we identified the following prompt engineering scheme as the most effective, occasionally producing camera views resembling the specified parameters. The focus was on roll, pitch, and vFoV, as distortion effects could not be replicated. To each prompt, we appended the following descriptions:

- Roll below 0°: Dutch angle shot, frame tilted  $\langle roll\ value \rangle$  degrees to the left.
- Roll above 0°: Dutch angle shot, frame tilted  $\langle roll\ value \rangle$  degrees to the right.
- Pitch below 0°: Picture taken with a high angle, bird’s view of  $\langle pitch\ value \rangle$ .
- Pitch above 0°: Picture taken with a low angle, worm’s view of  $\langle pitch\ value \rangle$ .
- VFoV below than 30°: Picture taken with a vertical field of view of  $\langle vFoV\ value \rangle$  degrees, a extreme close-up shot.
- VFoV between 30° and 55°: Picture taken with a vertical field of view of  $\langle vFoV\ value \rangle$  degrees, a close-up shot.
- VFoV between 55° and 75°: Picture taken with a vertical field of view of  $\langle vFoV\ value \rangle$  degrees, a medium shot.
- VFoV between 75° and 90°: Picture taken with a vertical field of view of  $\langle vFoV\ value \rangle$  degrees, a long shot.
- VFoV above 90°: Picture taken with a vertical field of view of  $\langle vFoV\ value \rangle$  degrees, a extreme long shot.

## E. User Study

The lack of reliable metrics to quantify the alignment between the input camera view and the camera view generated by our model prompted us to conduct a user study comparing our approach with existing models. In this study, participants evaluated and ranked the alignment of generated images with a specific camera view represented as PF-US. The comparison included images produced by three models: ours, SDXL, and Adobe Firefly. To generate the SDXL and Adobe Firefly images, we applied the prompt engineering detailed in Sec. ??, incorporating the appropriate tag for Adobe Firefly when available. A total of 34 participants ranked 16 examples after becoming familiar with the PF-US representation. Tab.1 presents the percentage of times each model was ranked first, second, or third, along with the rank product for each model. The results indicate that our model outperforms previous approaches, being most frequently selected as the best match for the desired camera views. Fig. 19- 20 illustrates the 16 examples used in the study, showing that SDXL rarely produces a camera view

similar to the target, while Adobe Firefly lacks the precision needed for accurate control.

Despite these insights, the user study has certain limitations as an evaluation metric. Some participants encountered difficulties in interpreting the PF-US representation, which may have affected their ability to accurately assess alignment.

Table 1. Percentage of times each model was ranked as the best, second-best, and third-best match to the target camera view. The final column presents the rank product, highlighting the superior performance of our model compared to SDXL and Adobe Firefly.

Model	1st best	2nd best	3rd best	Rank Product ↓
Adobe Firefly	14.89%	54.23%	30.88%	2.14
SDXL	4.04%	33.09%	62.87%	2.57
Ours	81.07%	12.68%	6.25%	<b>1.20</b>

## F. Generated Camera View Accuracy

The absence of reliable metrics to evaluate the camera parameters of generated images challenges the ability to assess the precision of our camera control. Jin et al. [24] offer a PF-estimator that given an image predicts its roll, pitch, and vFoV. However, the PF-estimator is not perfectly accurate (see Sec. B in the supplementary), thus we first assessed its general accuracy using our dataset, setting a baseline by estimating the camera parameters and calculating the median error between the estimation and the ground truth (see Tab. 2). Note that the PF-estimator does not predict  $\xi$ . Then, we estimated the PF of 1,432 images generated with our model using the PF-estimator. The images were generated with various prompts, varying each camera parameter in 5° increments. Table 2 shows the median error in degrees (difference between input and estimated parameters). Our results show that PreciseCam’s precision aligns with the accuracy achieved by the PF-estimator for the ground-truth data.

Table 2. PF-estimator accuracy as the median error between the ground truth and estimated parameters for the dataset (*baseline*), and between the input and estimated parameters for *our model*.

PF-Estimator Accuracy	Baseline	Our Model
Roll	8.65°	6.31°
Pitch	4.91°	12.01°
vFoV	16.88°	16.08°

## G. Consistent Camera for Input Variations

PreciseCam consistently generates the specified camera view regardless of variations in input noise or prompts. Fig. 16 shows our model’s ability to produce diverse image alternatives with the correct camera view when the input noise varies while keeping the prompt and camera parameters fixed. Additionally, Fig. 17 illustrates that changes in



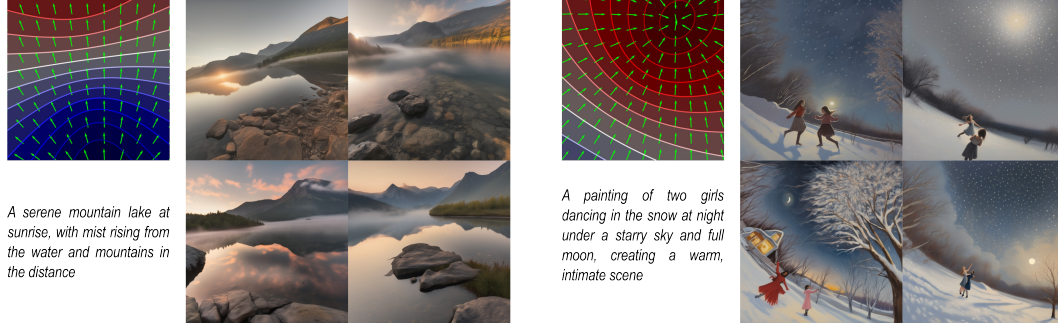


Figure 16. Generated images for different input noises but using the same prompt and camera parameters. PreciseCam produces different images while adhering to the specified camera parameters represented as the PF-US map.

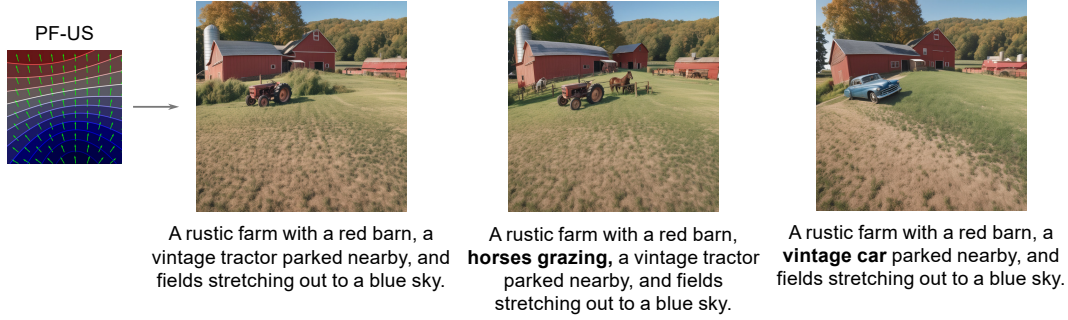


Figure 17. Generated images for small variations in prompt using the same noise and camera parameters. PreciseCam produces different images based on the prompt description while maintaining the specified camera view represented as the PF-US map.

the prompt do not affect the final camera view of the generated images for a given set of camera parameters and input noise. Instead, the model adjusts the content to align with the modified prompt.

## H. Compatibility with Multiple ControlNets

PreciseCam is compatible with other ControlNet models [54], such as pose, depth, or edge maps. As shown in Figure 18, our model integrates seamlessly with various ControlNets. While pose control<sup>2</sup> adjusts the subject’s pose, it does not control the background. By using PreciseCam, we apply camera view control to the final image while achieving the desired person’s pose. Additionally, in challenging cases where depth maps only represent objects without defining the background’s depth, our model can boost the generation of a coherent background with an accurate perspective<sup>3</sup>. Notice in Fig. 18 how the background perspective generated with PreciseCam aligns more closely with the house’s perspective.



Figure 18. PreciseCam is compatible with previous ControlNets, including pose control (left) and depth control (right). We showcase control over the person’s pose while simultaneously controlling the camera view, and our ability to generate images based on depth inputs while maintaining a background with consistent perspective. In the depth example, observe the change in perspectives of the red house in the background when we include camera control (right-bottom).

<sup>2</sup><https://huggingface.co/thibaud/controlnet-openpose-sdxl-1.0>

<sup>3</sup><https://huggingface.co/diffusers/controlnet-depth-sdxl-1.0>

## **I. Video Demo**

We provide a supplementary video highlighting the usability of our model. The video shows how users can intuitively adjust camera parameters with sliders to preview the desired camera view and generate an image based on the prompt.



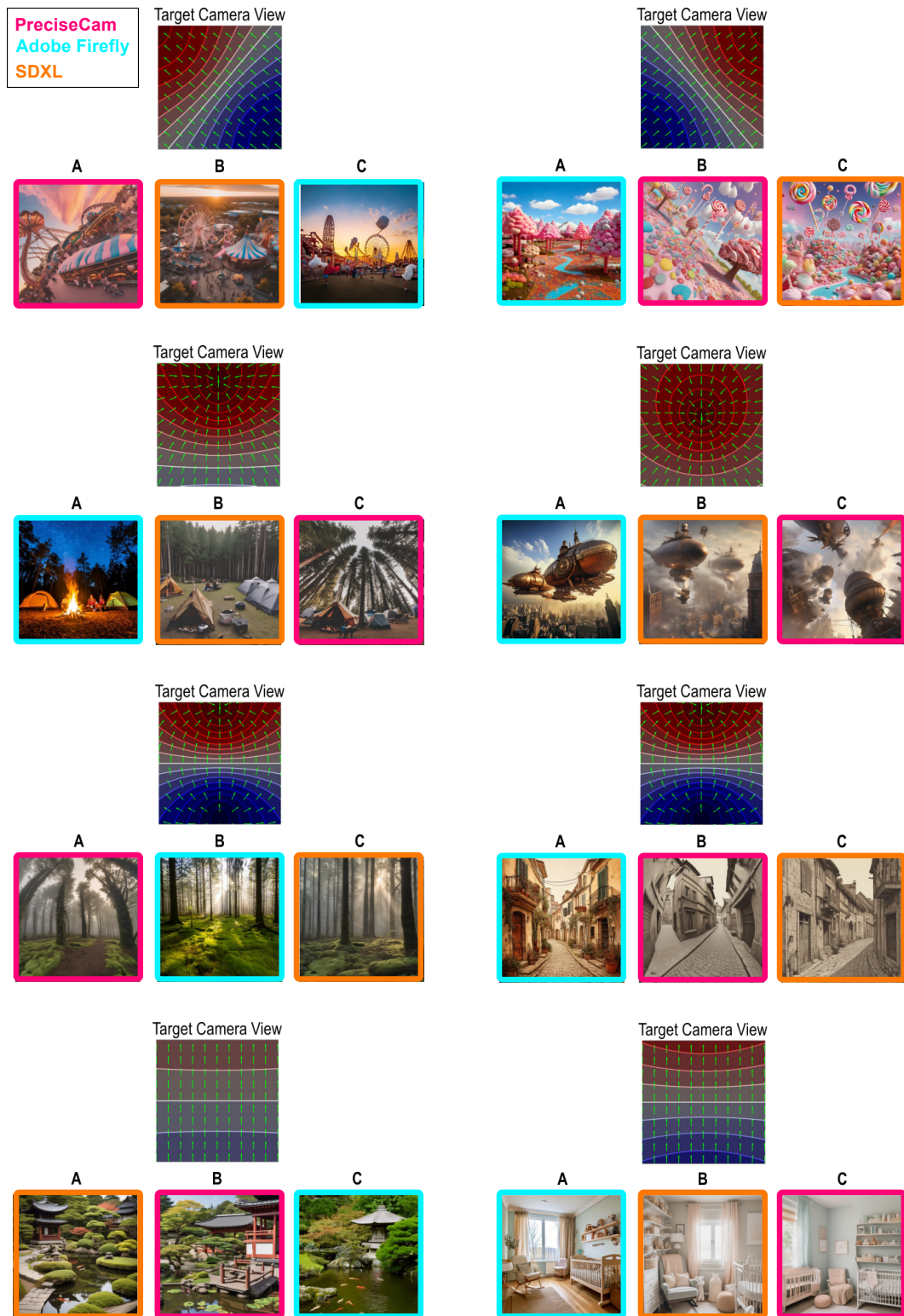


Figure 20. User Study Questions 1-8: Participants were shown sample images and asked to rate how well images A, B, and C matched the target camera view shown at the top. As the order was randomized, the color coding indicates whether the images correspond to our PreciseCam, Adobe Firefly, or SDXL.



## References

- [1] 360cities. 360cities dataset. <https://www.360cities.net/>. 5, 1
- [2] Adobe. Firefly. <https://www.adobe.com/products/firefly.html>. 1
- [3] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, et al. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 3
- [4] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. 2023. 3
- [5] Joao P Barreto. A unifying geometric representation for central projection systems. *Computer Vision and Image Understanding*, 2006. 2, 3
- [6] Shariq Farooq Bhat, Niloy Mitra, and Peter Wonka. Loosec-control: Lifting controlnet for generalized depth conditioning. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 5, 7
- [7] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 8
- [8] Oleksandr Bogdan, Viktor Eckstein, Francois Rameau, and Jean-Charles Bazin. Deepcalib: a deep learning approach for automatic intrinsic calibration of wide field-of-view cameras. In *Proceedings of the 15th ACM SIGGRAPH European Conference on Visual Media Production*, 2018. 3
- [9] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *ICCV*, pages 1511–1520, 2017. 2
- [10] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. Sketch2photo: Internet image montage. *ACM SIGGRAPH*, 2009. 2
- [11] Ta-Ying Cheng, Matheus Gadelha, Thibault Groueix, Matthew Fisher, Radomir Mech, Andrew Markham, and Niki Trigoni. Learning continuous 3D words for text-to-image generation. In *CVPR*, 2024. 3
- [12] Guillaume Couairon, Marlene Careil, Matthieu Cord, Stéphane Lathuilière, and Jakob Verbeek. Zero-shot spatial layout conditioning for text-to-image diffusion models. In *ICCV*, 2023. 3
- [13] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. *ACM SIGGRAPH*, 2001. 2
- [14] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *ICCV*, 1999. 2
- [15] Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. Photosketch: A sketch based image query and compositing system. In *ACM SIGGRAPH*, 2009. 2
- [16] Dave Epstein, Allan Jabri, Ben Poole, Alexei Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *NeurIPS*, 2023. 3
- [17] Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. <https://github.com/huggingface/accelerate>, 2022. 1
- [18] Poly Haven. HDRIs dataset. <https://polyhaven.com/hdris>. 5, 1
- [19] James Hays and Alexei A Efros. Scene completion using millions of photographs. *ACM SIGGRAPH*, 2007. 2
- [20] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024. 3, 8
- [21] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Cullers, and David H Salesin. Image analogies. *ACM SIGGRAPH*, 2001. 2
- [22] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipsecore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021. 6, 3
- [23] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2
- [24] Linyi Jin, Jianming Zhang, Yannick Hold-Geoffroy, Oliver Wang, Kevin Blackburn-Matzen, Matthew Sticha, and David F Fouhey. Perspective fields for single image camera calibration. In *CVPR*, 2023. 2, 3, 5, 1
- [25] Matthew Johnson, Gabriel J Brostow, Jamie Shotton, Ognjen Arandjelovic, Vivek Kwatra, and Roberto Cipolla. Semantic photo synthesis. In *Comput. Graph. Forum*, 2006. 2
- [26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, 2014. 1
- [27] Nupur Kumari, Grace Su, Richard Zhang, Taesung Park, Eli Shechtman, and Jun-Yan Zhu. Customizing text-to-image diffusion with camera viewpoint control. *arXiv preprint arXiv:2404.12333*, 2024. 3, 2
- [28] Jia Li, Jinming Su, Changqun Xia, and Yonghong Tian. Distortion-adaptive salient object detection in 360° omnidirectional images. *IEEE Journal of Selected Topics in Signal Processing*, 2019. 5, 1
- [29] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, 2023. 5, 2
- [30] Han Lin, Jaemin Cho, Abhay Zala, and Mohit Bansal. Ctrl-adapter: An efficient and versatile framework for adapting diverse controls to any diffusion model. *arXiv preprint arXiv:2404.09967*, 2024. 8
- [31] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2022. 3
- [32] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *AAAI*, 2024. 3
- [33] Semih Orhan and Yalin Bastanlar. Semantic segmentation of outdoor panoramic images. *Signal, Image and Video Processing*, 2022. 5, 1



- [34] Karran Pandey, Paul Guerrero, Metheus Gadelha, Yannick Hold-Geoffroy, Karan Singh, and Niloy J. Mitra. Diffusion handles: Enabling 3d edits for diffusion models by lifting activations to 3d. *CVPR*, 2024. 3
- [35] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 2
- [36] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 4
- [37] Anton Razzhigaev, Arseniy Shakhmatov, Anastasia Maltseva, Vladimir Arkhipkin, Igor Pavlov, Ilya Ryabov, Angelina Kuts, Alexander Panchenko, Andrey Kuznetsov, and Denis Dimitrov. Kandinsky: an improved text-to-image synthesis with image prior and latent diffusion, 2023. 2
- [38] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *CVPR*, 2021.
- [39] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 2022. 2
- [40] Vincent Sitzmann, Ana Serrano, Amy Pavel, Maneesh Agrawala, Diego Gutierrez, Belen Masia, and Gordon Wetstein. Saliency in VR: How do people explore virtual environments? *IEEE TVCG*, 2018. 5, 1, 2
- [41] Alexander Veicht, Paul-Edouard Sarlin, Philipp Lindenberger, and Marc Pollefeys. GeoCalib: Single-image Calibration with Geometric Optimization. In *ECCV*, 2024. 1
- [42] Andrey Voynov, Amir Hertz, Moab Arar, Shlomi Fruchter, and Daniel Cohen-Or. Curved diffusion: A generative model with optical geometry control. In *ECCV*, 2024. 3
- [43] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018. 2
- [44] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 2
- [45] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *NeurIPS*, 2024. 3, 8
- [46] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH*, 2024. 3, 8
- [47] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 2
- [48] DeJia Xu, Weili Nie, Chao Liu, Sifei Liu, Jan Kautz, Zhangyang Wang, and Arash Vahdat. Camco: Camera-controllable 3d-consistent image-to-video generation. *arXiv preprint arXiv:2406.02509*, 2024. 3, 8
- [49] Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory. *arXiv preprint arXiv:2308.08089*, 2023. 3, 8
- [50] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022. 2
- [51] Yu Zeng, Zhe Lin, Jianming Zhang, Qing Liu, John Collosse, Jason Kuen, and Vishal M Patel. Scenecomposer: Any-level semantic image synthesis. In *CVPR*, 2023. 3
- [52] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiao-gang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017. 2
- [53] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiao-gang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE TPAMI*, 2018. 2
- [54] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *CVPR*, 2023. 2, 3, 4, 8, 1
- [55] Yi Zhang, Lu Zhang, Wassim Hamidouche, and Olivier Deforges. A fixation-based 360 benchmark dataset for salient object detection. In *ICIP*, 2020. 5, 1